International Journal of Innovative Technology and Exploring Engineering (IJITEE)
ISSN: 2278-3075 (Online), Volume-9 Issue-2, December 2019

Deep Learning based Arrhythmia Classification with an ECG Acquisition System

Roshan Badrinath, Abhay Navada, Harshith Narahari, Ankit Datta, Dr. Sarojadevi H.

Abstract*: One of the issues that the human body faces is arrhythmia, a condition where the human heartbeat is either irregular, too slow or too fast. One of the ways to diagnose arrhythmia is by using ECG signals, the best diagnostic tool for detection of arrhythmia. This paper describes a deep learning approach to check whether signs of arrhythmia, in a given input signal, are present or not. A batch normalized CNN is used to classify the ECG signals based on the different types of arrhythmia. The model has achieved 96.39% training accuracy and 97% testing accuracy. The ECG signals are classified into five classes namely: Normal beats, Premature Ventricular Contraction (PVC) beats, Right Bundle Branch Block (RBBB) beats, Left Bundle Branch Block (LBBB) beats and Paced beats. A peak detection algorithm with six simple steps is designed to detect R-peaks from the ECG signals. A hardware device is built using Raspberry Pi to acquire ECG signals, which are then sent to the trained CNN for classification. The data-set for training is obtained from the MIT-BIH repository. Keras and Tensorflow libraries are used to design and develop the CNN and an application is designed using 'MEAN' stack and 'Flask' based servers.

***Keywords*: ECG Classification, Arrhythmia, Convolutional Neural Network, Batch Normalization, Peak Detection, IoT**

I. INTRODUCTION

Arrhythmia is the term that is used to describe abnormal heart beat. The heart may beat slower or faster, earlier or later, than usually seen. People may feel that their heart is beating too fast because the electrical signal which controls the heart is not working properly. Arrhythmia, in some cases can lead to stroke or cardiac arrest if the heart is weak or damaged. Electrocardiogram (ECG) is the tool used by cardiologists to detect arrhythmia. The process of acquiring the electrical activity of the heart, captured over time by external electrodes attached to the skin is known as Electrocardiography (ECG).

Revised Manuscript Received on December 30, 2019.

* Correspondence Author

Roshan Badrinath*, Bachelor of Engineering degree in Computer Science and Engineering from Nitte Meenakshi Institute of Technology (NMIT), Bengaluru, India.

Abhay Navada, Bachelor of Engineering degree in Computer Science and Engineering from NMIT, Bengaluru, India.

Harshith Narahari, Bachelor of Engineering degree in Computer Science and Engineering from NMIT, Bengaluru, India.

Ankit Datta, Bachelor of Engineering degree in Computer Science and Engineering from NMIT, Bengaluru, India.

Dr. Sarojadevi H., Professor, Department Of CSE at NMIT, Bengaluru, India.

The cell membranes of every cell in the outer layer of the heart have a charge associated with them, which depolarizes and re polarizes during each heartbeat. These appear as minute electrical signals at the skin, which can be amplified using the electrocardiography technique. Figure 1 shows an ideal ECG waveform consists of P, Q, R, S and T peaks.

The process of conducting an electrocardiogram test is easy, but it takes years of training for a human eye to correctly interpret the results. If irregularities in functioning of the heart and heart diseases can be detected early on in one's life, it provides the opportunity to prolong that person's life and improve his quality of living. As a result, to increase the accuracy and speed of obtaining the results, the automation of the process and classification of the ECG signals is necessary.

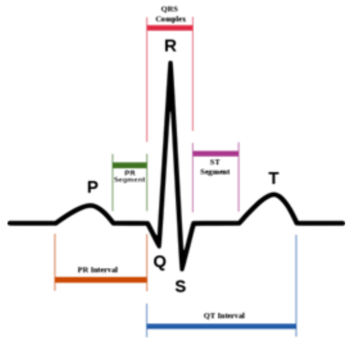

Fig. 1. A normal ECG signals showing P, Q, R, S and T peaks. [1]

This paper aims at classifying ECG signals with the purpose of finding out if a given person suffers from arrhythmia and identify the type of arrhythmia, in case it is detected, given the person's ECG signal. A deep learning approach using Convolutional Neural Networks (CNN) is used to classify the ECG signals. The data set used for training the CNN is obtained from the MIT- BIH database [2]. The database consists of 48 records of each slightly over 30 minutes long. There are 16 classes of signals present in the database out of which 5 classes, namely 'Normal', 'PVC', 'LBBB', 'RBBB' and 'Paced', are chosen in this implementation. Two CNNs designed for the classification. The difference between the two CNNs is that one of the CNN has Batch Normalization layers and the other has extra convolution layers. The results obtained by the CNNs are compared and the effect of adding Batch Normalization layer is evaluated. This paper also describes a Raspberry Pi based IoT device which has been built to acquire ECG signals. The acquired signals are then classified using the trained CNN.

An application is designed using MEAN stack which generates the result with a press a single button. The 'MEAN' stack consists of frameworks such as 'Angular JS' for the front-end, 'Express JS' for the back-end server, 'Node JS' as the run-time environment and 'MongoDB' database. Additionally, 'Flask' based servers are used as the classifier and in the Raspberry Pi board.

II. RELATED WORK

Over the last decade, many machine learning approaches have been applied to classify ECG signals. Machine Learning algorithms such as Support Vector Machines (SVM) [3, 4] and K-Nearest Neighbors (KNN) [4] have been used for the classification. One of the studies proposed the use of a deep feed-forward neural network [5] for classification which involved the extraction of six types of features from the ECG signal. The above techniques have shown high accuracies but the feature extraction is handcrafted.

There are many deep learning methods proposed in the last few years in which the feature extraction is done automatically by the neural networks. A recent study proposed the detection of myocardial infarction [6] from the PTB database. In this study, a deep 11-layered CNN is used for classification and it achieved an accuracy of 93.53%. Another study has proposed a CNN for the classification of ECG signals from the MIT-BIH and PTB databases [7]. Five classes in AAMI EC57 standard from the MIT-BIH database and two classes from the PTB database are used classification. The proposed CNN was trained separately and achieved an accuracy of 93.4% and 95.9% for MIT-BIH database classification and PTB database classification respectively. One of the studies proposed the use of 'AlexNet', a pre-trained deep CNN [8]. In this study, R-T segments of three classes from MIT-BIH database were used for classification. This approach has achieved a training accuracy of 98.51%. A combination of CNN and RNN (Recurrent Neural Network) have also been used for the classification of ECG signals. A recent study has proposed such a method where LSTM (Long Short Term Memory) and GRU (Gated Recurrent Unit) based recurrent layers have been attached to CNN [9] .

R-peak detection is an important step in pre-processing the data. Peak detection is necessary to generate images of each ECG waveform for training the CNN. A recent study has compared various peak detection methods such as Pan-Tompkins, Hilbert transform, Histogram approach, Wavelet transform, Auto-regression (AR), Independent Component Analysis (ICA), Linear prediction (LP), Adaptive threshold [10]. This study shows that Hilbert and Histogram methods have achieved the best results.

Many designs for IoT based ECG signal acquisition devices have been proposed. An Arduino board and Android based method is proposed in one of the papers to acquire ECG signals [11]. In this approach, 'Parse' is used as the back-end for the mobile application. Another paper has proposed an Arduino, Node JS and AWS (Amazon Web Service) based model [12]. Node JS back-end server is hosted on AWS EC2 instance which accepts requests from the Arduino hardware.

III. METHODOLOGY

A. Deep Learning Model

There are three stages involved in the process of training the deep learning model, namely R-peak detection, data selection and pre-processing, and designing the CNN architecture.

a. Peak Detection

The R-peak detection is required to generate images which will be used to train and test the CNN model. The CNN model is trained and tested using ECG signals from both the MIT-BIH database and the ECG acquisition system. The frequency of occurrence of these values are different for both the types. The same peak detection algorithm is used for both the data sets. Both the signals are represented as an array of ECG voltage values. The algorithm consists of six stages. The first step involves finding the difference between successive elements of the array. Then, the absolute values of all the elements are found. After that, the resultant array is subjected to Simple Moving Average (SMA). Since, SMA reduces the size of the array, the array is padded with zeroes in the beginning. Then, the resultant array is subjected to an activation where the mean of the array is the threshold. It is observed that the activation creates a plateau-like region near the QRS complex. Finally, the R-peaks are found to be the maximum value present in every plateau-like region. The complete process is visualized in Figure 2 for normal and PVC beats from the MIT-BIH database.

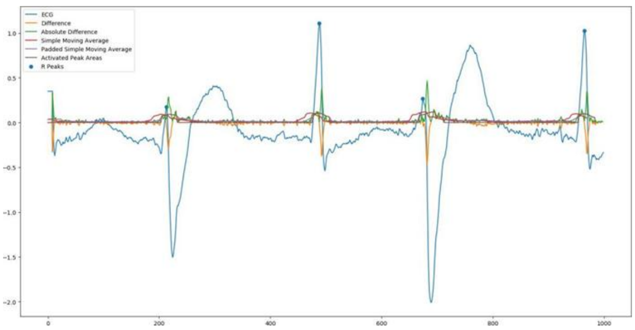

Fig. 2. The 6-stage R-peak detection algorithm. The blue dots in the figure represent the R-peaks.

b. Data Selection and Pre-processing

After R-peak detection, the next step is to prepare data for the CNN. During training, the signals are plotted and images are generated from the plot. During deployment of the application, the signals are plotted and buffered in the memory. The first 100 beats from the records numbered '106', '109', '118' and '217' in the MIT-BIH arrhythmia database, which represent the 'PVC', 'LBBB', 'RBBB' and 'Paced' beats classes respectively, are chosen for training, validation and testing purposes. For 'Normal' beats, the first 50 beats from record '100' of MIT-BIH arrhythmia database and 50 non-arrhythmic beats from ECG acquisition system are chosen to train the model. Images of R-T segments of each beat in the data set is generated. The R-peak detection algorithm described above is used to detect the R-peaks.

The T-peak is approximated to be present at a distance of 200 segments from the R-peak for signals from the MIT-BIT database and 70 segments from R-peak for signals from the ECG acquisition system respectively.

c. CNN Architectures

Two types of CNN architectures, with minor variations in the way they are designed but major variations in the speed at which they provide accurate results, were used. Figure 3 shows the architecture of the CNNs. The first CNN involves an extra Convolution layer after a Convolution layer followed by ReLU, MaxPool and Dropout layers. This sequence of layers is repeated for 3 times and followed by fully-connected layers. This architecture has shown a training and testing accuracies of 93.61% and 98% respectively. The second CNN involves using a Batch Normalization layer after each Convolution layer followed by ReLU activation, MaxPool and Dropout layers. This sequence of layers is repeated 3 times and then followed by fully-connected dense layers, as in the first CNN. This architecture has shown a training and testing accuracy of 96.39% and 97% respectively. Both the architectures were trained using categorical cross-entropy loss and Adam optimizer.

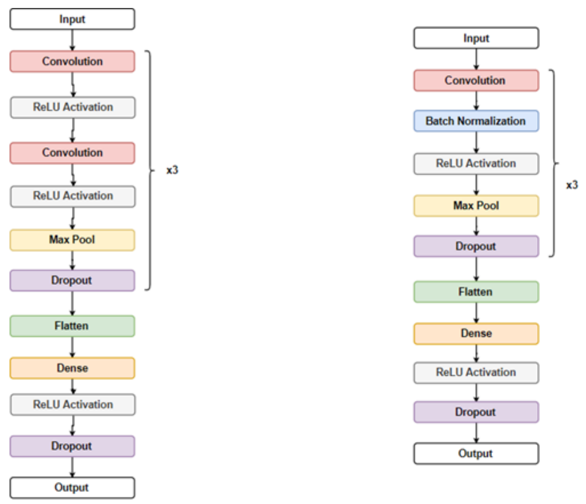

Fig. 3. Architecture of CNNs with (left) extra Convolution layers and (right) Batch Normalization layers.

B. System Architecture

A Raspberry Pi 3 board with modules AD8232 and ADS1115 is used for ECG signal acquisition. A 'Flask' server is used in the Raspberry Pi. By the click of a single button, a request to start the entire process is sent to ExpressJS back-end server. The ExpressJS Server sends a request to the Flask server running on the Raspberry Pi board. The Flask server acquires the ECG signal for 10 seconds. The voltage values are sent as a response back to the ExpressJS server. These voltage values are sent to the Flask classification server. The classification server generates images from the voltage values by plotting it. The buffered images are then sent to the classifier for classification. The result of the classification is then sent as a response back to the ExpressJS server. The ExpressJS server saves the result in the MongoDB database and sends a response back to the user. The complete process is shown in Figure 4.

IV. RESULTS AND ANALYSIS

As shown in Figure 3, there are two CNNs designed for the training process. The first CNN consists of extra convolution layers and the second CNN consists of batch normalization layers. The first CNN was trained for 100 epochs and the second CNN was trained for 30 epochs. As shown in Figure 5, the first and second CNN architectures have shown training accuracies of 93.61% and 96.39%, as well as testing accuracies of 98% and 97% respectively. The use of batch normalization layers has led the second CNN to converge faster than the first CNN. Also, the first CNN has 66% more parameters to train. When the size of training data is very small, having more trainable parameters might lead to over-fitting of the training data. Adding batch normalization layers prevents this from happening as the results from the previous layer will be constrained.

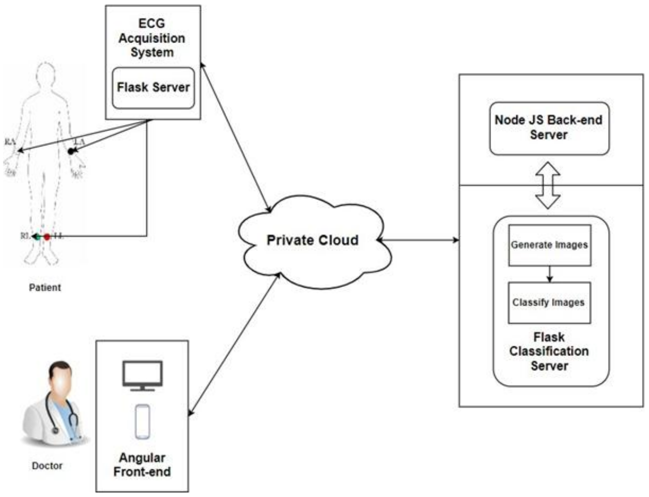

Fig. 4. The system architecture depicting an ECG acquisition system, client application and server connected to a private cloud.

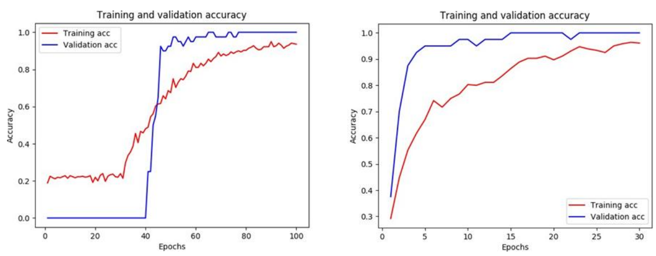

Fig. 5. Training graph of CNN with (left) extra Convolution layers and (right) Batch Normalization layers.

V. CONCLUSION

This paper aims at describing a procedure to design a CNN model to classify ECG signals based on the type of arrhythmia detected. Two CNN models were designed in which the first CNN had extra convolution layers and the second CNN had batch normalization layers instead.

The classifier was trained for 5 class labels namely, ’Normal’, ’PVC’, ’LBBB’, ’RBBB’ and ’Paced’. The classifier is developed using Keras and Tensorflow libraries. The first and second CNN architectures have shown training accuracies of 93.61% and 96.39%, as well as testing accuracies of 98% and 97% respectively. The CNN with batch normalization layers was chosen for deployment.

A 3-lead ECG acquisition device was built using Raspberry Pi, AD8232 ECG module and ADS1115 A-to-D converter. A Flask server was deployed on Raspberry Pi to acquire ECG signals for 10 seconds. An application which acted as an interface between the ECG acquisition device and the CNN classifier was deployed using MEAN stack. Additionally, a Flask server was used for classification of ECG signals. The IoT device begins to acquire ECG signals from the Angular application with the click of a single button.

Future implementations can aim at improving upon the model built by attempting to classify arrhythmia caused due to early beats. In addition, a real-time application which automatically keeps sending ECG data to the doctor can be built. Also, incorporating better noise removal from the hardware will help to improve the accuracy of the model.

REFERENCES

1. Electrocardiogram. URL: https://en.wikipedia.org/wiki/Electrocardiography
2. MIT data format. URL: https://www.physionet.org/physiobank/database/html/mitdbdir/records.htm
3. F. Melgani, Y. Bazi, Classification of Electrocardiogram Signals With Support Vector Machines and Particle Swarm Optimization, IEEE Transactions on Information Technology in Biomedicine, 2008.
4. O. Malgina, J. Milenkovic, E. Plesnik, M. J. F. Tasic, ECG signal feature extraction and classification based on R peaks detection in the phase space, in: IEEE Conference and Exhibition, IEEE, 2011.
5. T. J. Jun, H. J. Park, N. H. Minh, D. Y. Kim, Premature Ventricular Contraction Beat Detection with Deep Neural Networks, in: 15th IEEE International Conference on Machine Learning and Applications (ICMLA), IEEE, 2016.
6. U. R. Acharya, H. Fujita, S. L. Oh, Y. Hagiwara, J. H. Tan, M. Adam, Application of deep convolu- tional neural network for automated detection of myocardial infarction using ECG signals, Information Sciences.
7. M. Kachuee, S. Fazeli, M. Sarrafzadeh, ECG Heartbeat Classification: A Deep Transferable Represen- tation, in: 2018 IEEE International Conference on Healthcare Informatics (ICHI), IEEE, 2018.
8. S. Ozdalili, A. Isin, Cardiac arrhythmia detection using deep learning, in: 9th International Conference on Theory and Application of Soft Computing, Computing with Words and Perception (ICSCCW), Vol. 120, Elsevier, 2017, pp. 268–275.
9. G. Swapna, K. P. Soman, R. Vinayakumar, Automated detection of cardiac arrhythmia using deep learning, in: International Conference on Computational Intelligence and Data Science (ICCIDS 2018), Elsevier, 2018.
10. A. Agrawal, D. H. Gawali, Comparative study of ECG feature extraction methods, in: 2017 2nd IEEE International Conference On Recent Trends in Electronics Information; Communication Technology (RTEICT), IEEE, 2017.
11. M. Machado, E. G. Bertogna, M. A. Sovierzoski, An IoT Approach to an ECG Online Monitor System in an Android Application, in: VII Latin American Congress on Biomedical Engineering (CLAIB 2016), Springer, 2017, pp. 237–240.
12. I. Villanueva-Miranda, H. Nazeran, R. Martinek, CardiaQloud: A Remote ECG Monitoring System Us- ing Cloud Services for eHealth and mHealth Applications, in: 2018 IEEE 20th International Conference on e-Health Networking, Applications and Services (Healthcom), IEEE, 2018.

AUTHORS PROFILE

Roshan Badrinath has completed his Bachelor of Engineering degree in Computer Science and Engineering from Nitte Meenakshi Institute of Technology (NMIT), Bengaluru, India.

Abhay Navada has completed his Bachelor of Engineering degree in Computer Science and Engineering from NMIT, Bengaluru, India. He has currently been admitted into the Master of Science program in Computer Science degree at the University of Southern California, Los Angeles, CA, USA.

Harshith Narahari has completed his Bachelor of Engineering degree in Computer Science and Engineering from NMIT, Bengaluru. He is currently working at Mindtree as a Software Engineer.

Ankit Datta has completed his Bachelor of Engineering degree in Computer Science and Engineering from NMIT, Bengaluru. He is currently working at Infosys as a Systems Engineer.

Dr. Sarojadevi H. is a professor in the department of CSE at NMIT, Bengaluru, India. She is a PhD graduate from the Indian Institute of Science, Bengaluru. She has received B.E. and M.E. qualifications from the University Visvesvaraya College of Engineering, Bangalore. She has published nearly 100 papers, 2 books and also patent. Her work experience of 23 years includes industry and academics. She has won several national and international recognitions, including Marquis Who’s Who in the World 2016.